REMARKS/ARGUMENTS

Amendments were made to the specification to correct errors and to clarify the specification. No new matter has been added by any of the amendments to the specification.

Claims 1-29 are pending in the present application. By this response, claims 1, 8, 10, 15, 22, 24, and 29 are amended. Claims 1, 15, and 29 are amended to clarify the subject matter that is being claimed. Support for the amendments may be found at least on page 30, lines 11-15. Claims 8, 10, 22, and 24 are amended in view of the amendments to claims 1 and 15 in order to provide proper antecedent basis. Reconsideration of the claims in view of the above amendments and the following remarks is respectfully requested.

I. Examiner Interview

Applicants thank Examiner Lai and Examiner Fleming for the courtesies extended to Applicants' representative during the July 12, 2006 telephone interview. During the interview, the objection to the drawings was discussed. Applicants' representative pointed to the location in the specification where element 260 was discussed. Also, during the interview, suggestions to amend the present application to overcome the 35 U.S.C. § 101 rejections were discussed. Claim 15 is amended to recite "A computer program product in a recordable-type computer readable medium..." The Examiners stated these amendments would overcome the 35 U.S.C. § 101 rejection. Also during the interview, proposed amendments to claims 1, 15, and 29 were discussed. Examiner Lai agreed that the Burrows reference teaches counting every event and not just the first event. Examiner Lai stated he would consider the proposed amendments. The substance of the interview is summarized in the remarks of sections that follow.

II. Drawings

The Office objects to the Drawings as including reference character(s) not mentioned in the description, specifically, element 260 of Figure 2. Applicants respectfully direct the Office to page 24, lines 12 and 15, and page 25, line 3 of the specification where element 260 is mentioned. In view of the above, Applicants respectfully request the objection to the Drawings be withdrawn.

III. Objection to Claims

The Office states that claims 15-26 and 29 were objected for minor informalities. In response, the claims have been rewritten to overcome these objections.

IV. 35 U.S.C. § 112, Second Paragraph

The Office rejects claim 29 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter, which applicants regard as the invention. By this response, claim 29 is amended to recite "An apparatus for processing instructions, the apparatus comprising." Therefore, Applicants request withdrawal of the rejection of claim 29 under 35 U.S.C. § 112, second paragraph.

V. 35 U.S.C. § 101

The Office rejects claims 3, 7, 17, and 21 under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. Applicants respectfully submit that claims 3, 7, 17, and 21 depend from claims 1 and 15. Since claims 1 and 15 are not rejected under 35 U.S.C. § 101 as having no tangible end results, Applicants respectfully submit claims 1 and 15 have tangible end results and Applicants respectfully submit that the subject matter recited in claims 3, 7, 17, and 21 also have practical application and are in the technical arts. Therefore, Applicants submit that claims 3, 7, 17, and 21 are statutory and Applicants respectfully request the withdrawal of the rejection of claims 3, 7, 17, and 21 under 35 U.S.C. § 101.

The Office rejects claims 15-28 under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. By this response, claim 15 is amended to recite "A computer program product in a recordable-type computer readable medium..." Therefore, Applicants respectfully submit that independent claim 15 is statutory. Thus, Applicants respectfully request withdrawal of the rejection of claims 15-28 under 35 U.S.C. § 101.

VI. 35 U.S.C. § 103, Obviousness

The Office rejects claims 1-29 under 35 U.S.C. § 103(a) as being unpatentable over Burrows (U.S. Patent No. 5,887,159) in view of Holmberg (U.S. Patent No. 6,233,679 B1). This rejection is respectfully traversed.

As to claim 1, the Office states:

> As per claim 1, Burrows teaches a method in a data processing system for processing instructions, the method comprising:
>
> responsive to receiving an instruction at a processor in the data processing system (See column 2, lines 49-59: Instructions are fetched, decoded and executed), determining whether an indicator is associated with the instruction (See column 2, lines 59-65: This is done by checking to see if hint information is null or not);
>
> enabling counting, by the processor, of each first event associated with a primary metric of the execution (See column 5, lines 11-13: The metric given is the number of times an execution flow is encountered) of the instruction if the indicator is associated with the instruction (See figure 5 and column 5, lines 11-13: A count field is available for keeping track of the number of times a certain action occurs), wherein the processor autonomically increments the count of the first events associated with the primary metric of the execution of the instruction in a first hardware counter (A counter inherently is able to increment a count when certain operations occurs);
>
> determining if the count of the first events associated with the primary metric of the execution of the instruction stored in the first hardware counter satisfies a predetermined relationship with a threshold value (See column 5, lines 14-17: Counts are tracked and used to update hint information);
>
> Burrows does not teach a second counter.
>
> Holmberg does teach a second counter (See figure 2) and enabling counting, by the processor, of each second event associated with a secondary metric of the execution of a portion of code associated with the instruction (See column 4, line 54 – column 5, line 2: A second counter is available for counter a separate event), wherein the processor autonomically increments the count of the second events associated with the secondary metric of the execution of a portion of code associated with the instruction in a second hardware counter (A counter inherently is able to increment a count when certain operations occurs).
>
> It would have been obvious to a person having ordinary skill in the art at the time the invention was made to modify the teachings of Burrows with Holmberg because utilizing a second counter would allow greater ability to track data and thus better predictions can be made. More data means more informed decisions can be made and thus accuracy is likely to be increased. The performance indicator is another form of providing more data. It is also of note that one of the references cited in Holmberg is Burrows.

Office Action dated May 5, 2006, pages 5-6.

Amended claim 1, which is representative of the other rejected independent claims 15 and 29 with respect to similarly recited subject matter, reads as follows:

> 1. A method in a data processing system for processing instructions, the method comprising:
>
> responsive to receiving an instruction at a processor in the data processing system, determining whether an indicator is associated with the instruction, **wherein the indicator identifies the instruction as one that is to be monitored by a performance monitor unit**;
>
> **enabling counting, by the processor, of each first event associated with a primary metric of the execution of the instruction if the indicator is associated with the instruction**, wherein the processor autonomically increments the count of the first

events associated with the primary metric of the execution of the instruction in a first hardware counter;

determining if the count of the first events associated with the primary metric of the execution of the instruction stored in the first hardware counter satisfies a predetermined relationship with a threshold value; and

enabling counting, by the processor, of each second event associated with a secondary metric of the execution of a portion of code associated with the instruction, wherein the processor autonomically increments the count of the second events associated with the secondary metric of the execution of a portion of code associated with the instruction in a second hardware counter. (emphasis added)

Burrows and Holmberg, taken alone or in combination, fail to teach or suggest responsive to receiving an instruction at a processor in the data processing system, determining whether an indicator is associated with the instruction, wherein the indicator identifies the instruction as one that is to be monitored by a performance monitor unit; enabling counting, by the processor, of each first event associated with a primary metric of the execution of the instruction if the indicator is associated with the instruction, wherein the processor autonomically increments the count of the first events associated with the primary metric of the execution of the instruction in a first hardware counter; determining if the count of the first events associated with the primary metric of the execution of the instruction stored in the first hardware counter satisfies a predetermined relationship with a threshold value; and enabling counting, by the processor, of each second event associated with a secondary metric of the execution of a portion of code associated with the instruction, wherein the processor autonomically increments the count of the second events associated with the secondary metric of the execution of a portion of code associated with the instruction in a second hardware counter.

Burrows is directed to locating hint fields embedded within instructions and replacing the hint fields with calls to intercept the execution flow and redirect to procedures of a monitor. During code execution, Burrows records the existing hint information in a memory and analyzes the hint information to determine the most frequently occurring or best__hint value. When a best__hint value has been determined, Burrows restores the replaced instructions with best__ hint values. (see Burrows, Abstract)

The Office alleges that Burrows teaches responsive to receiving an instruction at a processor in the data processing system, determining whether an indicator is associated with the instruction in the following section:

> FIG. 1 shows a process 100 which can be used for dynamically determining hint fields of instructions of machine executable code. A programmer generates, for example, object-oriented source programs 110 using conventional programming techniques. The source programs 110, once processed, are intended for execution in a computer system (CPU) 190. The programs 110 can be compiled by a compiler 111 into object code modules (obj) 120. The object code modules 120 include instructions with hint fields. Hint fields help branch prediction logic of the CPU 190 to determine the address of a next instruction to be fetched. Execution cycles are saved if the instructions are correctly

fetched. Instructions whose destination addresses can only be resolved at run-time have their hint fields set to null.

A monitor program 130 is also generated. The purpose of the monitor program 130 is to dynamically intercept the execution flow of the object modules 120 to record and analyze hint information.

(Burrows, column 2, lines 49-65)

In this section, Burrows describes determining which instructions have hint fields. The Office seems to equate Burrows' hint field to the presently claimed indicator. Burrows' hint fields are described as fields that help branch prediction logic of the CPU to determine the address of a next instruction to be fetched. Thus, Burrows' hint field specifies a likely target address. In contradistinction, the present invention in claim 1 provides an indicator that identifies the instruction as one that is to be monitored by a performance monitor unit. Therefore, Burrows does not teach or suggest determining whether an indicator is associated with the instruction in response to receiving an instruction at a processor in the data processing system, wherein the **indicator identifies the instruction as one that is to be monitored by a performance monitor unit**, as recited in claim 1.

Additionally, The Office alleges that Burrows teaches enabling counting, by the processor, of each first event associated with a primary metric of the execution of the instruction if the indicator is associated with the instruction, wherein the processor autonomically increments the count of the first events associated with the primary metric of the execution of the instruction in a first hardware counter in the following section:

The count field 530 indicates the number of times the execution flow has been intercepted.

(Burrows, column 5, lines 11-13)

In this section, Burrows describes a count field that indicates the number of times the execution of the code has been intercepted. Thus, Burrows increments the counter on a first interception, a second interception, third interception, etc. no matter what type of event or the number of times an event happens. In contradistinction, the present invention increments a counter on a **first event associated with a primary metric of the execution of the instruction** and **if the indicator is associated with the instruction**. Therefore, Burrows teaches away from the present invention by counting every time the execution code has been intercepted and Burrows does not teach or suggest enabling counting, by the processor, of each first event associated with a primary metric of the execution of the instruction if the indicator is associated with the instruction, wherein the processor autonomically increments the count of the first events associated with the primary metric of the execution of the instruction in a first hardware counter.

Furthermore, the Office alleges that Burrows teaches determining if the count of the first events associated with the primary metric of the execution of the instruction stored in the first hardware counter satisfies a predetermined relationship with a threshold value in the following section:

> The hit field 520 can be incremented and decremented. The best__hint field 540 yields a value which can be used in the hint field 220 of the replaced instruction. The value 540 is dynamically determined as the machine code is executing.

(Burrows, column 5, lines 14-17)

In this section, Burrows describes a counter for a hint field that can be incremented or decremented and a best__hint field that is a most frequently occurring hint field (see Burrows, Abstract). Burrows describes a hint field as fields that help branch prediction logic of the CPU to determine the address of a next instruction to be fetched. Thus, Burrows' hint field specifies a likely target address. Applicants respectfully submit that Burrows teaches that all of the hint fields associated with an address are counted in order to establish a best__hint field, so that the best__hint field may replace the hint field in the instruction. Thus, Burrows counts all of the hint fields and not just the first events. Consequently, Burrows teaches away from the present invention by counting every hint field and Burrows does not teach or suggest determining if the **count of the first events** associated with the primary metric of the execution of the instruction stored in the first hardware counter satisfies a predetermined relationship with a threshold value.

The Office acknowledges that Burrows does not teach or suggest enabling counting, by the processor, of each second event associated with a secondary metric of the execution of a portion of code associated with the instruction, wherein the processor autonomically increments the count of the second events associated with the secondary metric of the execution of a portion of code associated with the instruction in a second hardware counter. However, the Office alleges that Holmberg teaches this feature in the following section:

> In FIG. 2 the hardware used in the unit 121 for collecting statistics regarding if a branch is taken or not, is shown. Thus, for collecting statistics regarding a certain conditional branch instruction in the program memory, the address of that instruction is placed in a register 201, here termed Measured Address Register (MAR). This address is compared in a block 203 with the instruction address currently pointed to by the program counter and which is available in a block 205.
>
> The two addresses are compared in the block 203 and if the two addresses are identical a first counter in a block 211 is incremented by one. The output from the block 203 is also fed to an AND block 207. To the AND block 207, a signal indicating if the branch was taken or not is also fed. Thus, the output from the block 207 increments a second counter 209 each time the branch in the instruction in the memory address register is taken.
>
> In general, two out of the following statistics counts needs to be collected for setting the branch prediction bits:
>
> the number of times the conditional branch is taken

the number of times the conditional branch is not taken
the total number of times the conditional branch instruction is executed.

(Holmberg, column 4, line 47, to column 5, line 2)

In this section, Holmberg uses two different counters to count two of: the number of times the conditional branch is taken, the number of times the conditional branch is not taken, or the total number of times the conditional branch instruction is executed. Thus, once Holmberg encounters the first conditional jump instruction, it is loaded into a measured address register. Then, Holmberg increments counter each time the program from which statistics are collected executes the conditional branch instruction associated with the address stored in the MAR and when the corresponding branch is taken. Thus, Holmberg counts all of the times the conditional jump instruction is taken and the second event. Therefore, Holmberg does not teach or suggest enabling counting, by the processor, of **each second event** associated with a secondary metric of the execution of a portion of code associated with the instruction, wherein the processor autonomically increments the count of the second events associated with the secondary metric of the execution of a portion of code associated with the instruction in a second hardware counter.

The Office bears the burden of establishing a *prima facie* case of obviousness based on the prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). Since the references fail to teach or suggest responsive to receiving an instruction at a processor in the data processing system, determining whether an indicator is associated with the instruction, wherein the indicator identifies the instruction as one that is to be monitored by a performance monitor unit; enabling counting, by the processor, of each first event associated with a primary metric of the execution of the instruction if the indicator is associated with the instruction, wherein the processor autonomically increments the count of the first events associated with the primary metric of the execution of the instruction in a first hardware counter; determining if the count of the first events associated with the primary metric of the execution of the instruction stored in the first hardware counter satisfies a predetermined relationship with a threshold value; and enabling counting, by the processor, of each second event associated with a secondary metric of the execution of a portion of code associated with the instruction, wherein the processor autonomically increments the count of the second events associated with the secondary metric of the execution of a portion of code associated with the instruction in a second hardware counter, the Office has failed to establish a *prima facie* case of obviousness, because the Office does not show where each and every claim limitation is taught or fairly suggested by the applied prior art.

The applied references do not teach or suggest each and every claim limitation; therefore, Burrows and Holmberg, taken alone or in combination, do not render claim 1 obvious. Independent claims 15 and 29 recite similar subject matter addressed above with respect to claim 1 and are allowable for similar reasons. Since claims 2-14 and 16-28 depend from claims 1 and 15, the same distinctions

between Burrows and Holmberg and the invention recited in claims 1, 15, and 29 apply for these claims. Additionally, claims 2-14 and 16-28 recite other additional combinations of features not taught or suggested by the references.

Furthermore, no suggestion is present in any of the references to modify the references to include such features. That is, there is no teaching or suggestion in Burrows and Holmberg that a problem exists for which responsive to receiving an instruction at a processor in the data processing system, determining whether an indicator is associated with the instruction, wherein the indicator identifies the instruction as one that is to be monitored by a performance monitor unit; enabling counting, by the processor, of each first event associated with a primary metric of the execution of the instruction if the indicator is associated with the instruction, wherein the processor autonomically increments the count of the first events associated with the primary metric of the execution of the instruction in a first hardware counter; determining if the count of the first events associated with the primary metric of the execution of the instruction stored in the first hardware counter satisfies a predetermined relationship with a threshold value; and enabling counting, by the processor, of each second event associated with a secondary metric of the execution of a portion of code associated with the instruction, wherein the processor autonomically increments the count of the second events associated with the secondary metric of the execution of a portion of code associated with the instruction in a second hardware counter, is a solution. To the contrary, Burrows appears to teach counting every hint field and Holmberg appears to teach counting all of the times the conditional jump instruction is taken.

Moreover, neither reference teaches or suggests the desirability of incorporating the subject matter of the other reference. That is, there is no motivation offered in either reference for the alleged combination. The Office alleges that the motivation would be "utilizing a second counter would allow greater ability to track and thus better predictions can be made." The present invention provides for counting of each first event associated with a primary metric of the execution of the instruction if the indicator is associated with the instruction and counting of each second event associated with a secondary metric of the execution of a portion of code associated with the instruction. As discussed above, Burrows appears to teach counting every hint field and Holmberg appears to teach counting all of the times the conditional jump instruction is taken. Neither reference teaches or suggests counting each first event associated with a primary metric of the execution of the instruction if the indicator is associated with the instruction and counting each second event associated with a secondary metric of the execution of a portion of code associated with the instruction. Thus, the only teaching or suggestion to even attempt the alleged combination is based on a prior knowledge of Applicants' claimed invention thereby constituting impermissible hindsight reconstruction using Applicants' own disclosure as a guide.

One of ordinary skill in the art, being presented only with Burrows and Holmberg, and without having a prior knowledge of Applicants' claimed invention, would not have found it obvious to combine and modify Burrows and Holmberg to arrive at Applicants' claimed invention, as recited in claim 1. To the contrary, even if one were somehow motivated to combine Burrows and Holmberg, and it were somehow possible to combine the systems, the result would not be the invention, as recited in claim 1. The resulting system would still fail to count each first event associated with a primary metric of the execution of the instruction if the indicator is associated with the instruction and count each second event associated with a secondary metric of the execution of a portion of code associated with the instruction.

In view of the above, Applicants respectfully submit that the Burrows and Holmberg, taken alone or in combination, fail to teach or suggest the features of claims 1, 15, and 29. At least by virtue of their dependency on claims 1, 15, and 29, the features of dependent claims 2-14 and 16-28 are not taught or suggested by Burrows and Holmberg, whether taken individually or in combination. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 1-29 under 35 U.S.C. § 103.

Moreover, in addition to their dependency from independent claims 1 and 15, the specific features recited in dependent claims 2-14 and 16-28 are not taught by Burrows and Holmberg, either alone or in combination. For example, with regard to claims 3 and 17, Burrows and Holmberg, taken alone or in combination, do not teach or suggest wherein the indicator is stored in a performance instrumentation shadow cache and wherein the processor checks the performance instrumentation shadow cache to determine whether the indicator is associated with the instructions. The Office alleges that Burrows teaches these features in the following Figure and section:

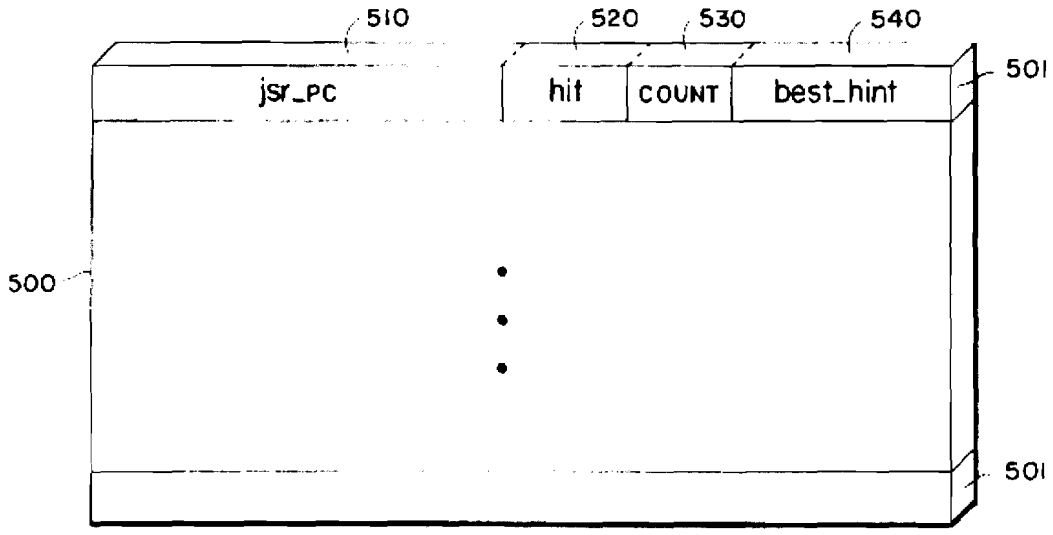

FIG. 5

(Burrows, Figure 5)

Instructions whose destination addresses can only be resolved at run-time have their hint fields set to null.

A monitor program 130 is also generated. The purpose of the monitor program 130 is to dynamically intercept the execution flow of the object modules 120 to record and analyze hint information.

(Burrows, column 2, lines 59-65)

While Burrows describes a hint prediction table in Figure 5, the table described by Burrows is created through the analysis of the instructions that are intercepted during the execution of the code. In column 2, lines 59-65, Burrows describes the interception of the instructions in order to read their hint fields. Thus, Burrows could not perform the steps of checking the performance instrumentation shadow cache to determine whether the indicator is associated with the instructions if the table described by Burrows is created after the instructions are intercepted.

Additionally, with regard to claims 5 and 19, Burrows and Holmberg, taken alone or in combination, do not teach or suggest wherein the indicator is a separate instruction. The Office alleges that Burrows teaches this feature in Figure 5, shown above, and in the following figure and section:

For example, if the instruction 171 to be replaced is of the form:
jsr Ra, Rb, 0.
e.g., target in Rb, and no hint, then the replacement instruction will be of the form:
bsr Ra, Rb__proc
where Rb__proc is the label of the monitor procedure for indirect calls through register Rb.

(Burrows, column 4, lines 18-25)

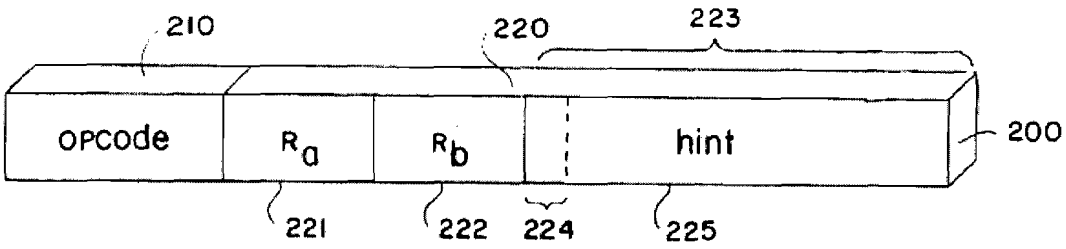

FIG.2

(Burrows, Figure 2)

In this section and Figures, Burrows describes a branch type instruction 200 that includes a hint field 223. Thus, Burrows teaches that a hint field is part of an instruction and not a separate instruction.

Further, with regards to claims 6 and 20, Burrows and Holmberg, taken alone or in combination, do not teach or suggest wherein the first events include at least one of an entry into a module, an exit from a module, an entry into a subroutine, an exit from a subroutine, an entry into a function, an exit from a

function, a start of input/output, a completion of input/output, and the execution of the instruction. As discussed above, Burrows teaches a counter for a hint field that can be incremented or decremented and a best__hint field that is a most frequently occurring hint field (see Burrows, Abstract). Burrows describes a hint field as fields that help branch prediction logic of the CPU to determine the address of a next instruction to be fetched. Thus, Burrows' hint field specifies a likely target address. Applicants respectfully submit that in Burrows all of the hint fields associated with an address are counted in order to establish a best__hint field so that the best__hint field may replace the hint field in the instruction. Thus, Burrows counts all of the hint fields and not just the first events. Burrows is simply not concerned with anything but counting the target addresses in order to improve jump condition accuracy. Applicants respectfully submit that hint field (a likely target address) is not an entry into a module, an exit from a module, an entry into a subroutine, an exit from a subroutine, an entry into a function, an exit from a function, a start of input/output, a completion of input/output, or the execution of the instruction.

Still further, with regard to claims 9 and 23, Burrows and Holmberg, taken alone or in combination, does not teach or suggest wherein the first hardware counter is a combined counter value hardware counter that stores a combined count from a plurality of other hardware counters. The Office alleges that this feature is taught by Burrows in Figure 5 and column 5, lines 11-13, shown above and by Holmberg at column 4, line 64 to column 5, line 2, shown above. Burrows only teaches one counter that counts the number of times a hint field is intercepted. Holmberg teaches the use of two different counters to count two of:

- the number of times the conditional branch is taken,
- the number of times the conditional branch is not taken, or
- the total number of times the conditional branch instruction is executed.

Thus, Holmberg counts either

- the number of times the branch is taken **and** the number of times the branch in not taken,
- the number of time the branch is taken **and** the total number of times the conditional branch instruction is executed, or
- the number of times the branch is not taken **and** the total number of times the conditional branch instruction is executed.

There is no teach or suggesting in Holmberg, that the number of times the branch is taken and the number of times the branch in not taken are combined to create the total number of times the conditional branch instruction is executed. Moreover, Holmberg only provides for two counters so that a comparison may be made between the counters. Thus, Burrows and Holmberg, taken alone or in combination, fail to teach or suggest wherein the first hardware counter is a combined counter value hardware counter that stores a combined count from a plurality of other hardware counters.

Thus, in addition to being dependent on independent claims 1 and 15, the specific features of dependent claims 2-14 and 16-28 are also distinguishable over Burrows and Holmberg, either alone or in combination, by virtue of the specific features recited in these claims. Accordingly, Applicants respectfully request withdrawal of the rejection of dependent claims 2-14 and 16-28 under 35 U.S.C. § 103

VII. Conclusion

It is respectfully urged that the subject application is patentable over the prior art of record and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: July 26, 2006

Francis Lammes
Reg. No. 55,353
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Agent for Applicants